

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2001-188702
(P2001-188702A)

(43) 公開日 平成13年7月10日 (2001.7.10)

(51) Int.Cl.
G 0 6 F 12/00

識別記号
5 4 5
5 2 0

F I
G 0 6 F 12/00

テームコード* (参考)

5 4 5 B 5 B 0 8 2
5 2 0 E

審査請求 未請求 請求項の数 5 O L (全 13 頁)

(21) 出願番号 特願平11-373771
(22) 出願日 平成11年12月28日 (1999. 12. 28)

(71) 出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目 6 番地
(72) 発明者 沼野井 淳
神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア事業部内
(72) 発明者 横塚 大典
神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア事業部内
(74) 代理人 100078134
弁理士 武 顕次郎

最終頁に続く

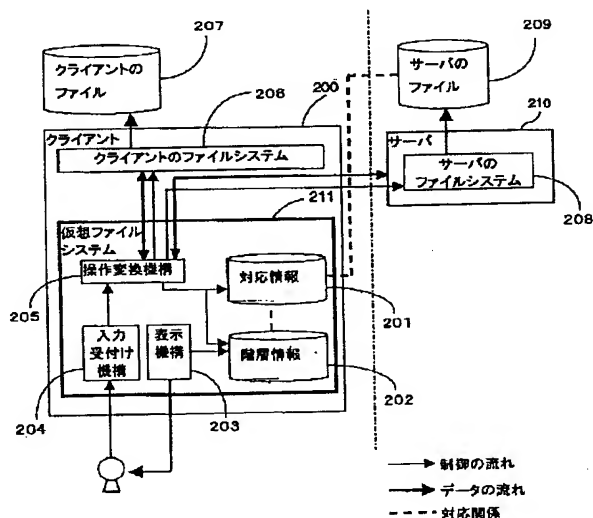
(54) 【発明の名称】 分散ファイルシステム及びファイル操作方法

(57) 【要約】

【課題】 概念が異なるファイルシステムを持つマシン相互間でクライアントのファイルとサーバのファイルとの両者を同様に扱うことができるようにし、作業効率を向上させ、操作ミスを減少させること。

【解決手段】 サーバのファイルを仮想的にクライアントのファイルとして取り扱うために便宜的な名称づけを行い、その便宜的な名称同士の階層構造を定義する階層情報 202 と、便宜的な名称とサーバの実ファイル 209 との対応関係を記述する対応情報 201 を用意する。対応情報 201 と階層情報 202 とを参照してファイルの情報を表示する表示機構 203 と、サーバのファイルに対して、クライアントのファイルシステムと同じ操作を受け付ける入力受け付け機構 204 と、その操作をサーバのファイルシステムに対する操作に変換する操作変換機構 205 とを備える。

図 1



【特許請求の範囲】

【請求項 1】 サーバのファイルシステムと異なるファイルシステムを持つクライアントとサーバとをネットワークにより接続して構成される分散ファイルシステムにおいて、前記クライアントは、前記サーバが持つ階層構造を持たないデータにより構成されるファイル内のデータを、階層構造により表わした階層情報として管理する階層情報保持手段と、該階層情報保持手段内の情報と前記サーバ内のファイルのデータとの対応情報を保持する対応情報保持手段と、サーバ内のファイルに対する操作が指示されたとき、前記階層情報及び対応情報に基づいて前記サーバのファイルをサーバのファイルシステムを介して操作する操作変換機構とを有する仮想ファイルシステムを備えて構成されることを特徴とする分散ファイルシステム。

【請求項 2】 前記仮想ファイルシステムは、前記操作変換機構がサーバ側ファイルシステムに対する操作を行う際に必要な情報が前記対応情報及び前記階層情報に記述されていない場合、サーバまたはクライアント内を探索することによって前記情報を補完する補完機構をさらに有することを特徴とする請求項 1 記載の分散ファイルシステム。

【請求項 3】 前記クライアントは、表示機構を備え、該表示装置にクライアントの仮想ファイルシステムが持つサーバのファイルの対応情報を、自クライアントのファイルシステムのファイルと同一の形式で表示させ、クライアントのファイルシステムからサーバのファイルを自クライアントのファイルを操作する方法と同一の方法により操作することを特徴とする請求項 1 または 2 記載の分散ファイルシステム。

【請求項 4】 サーバのファイルシステムと異なるファイルシステムを持つクライアントとサーバとをネットワークにより接続して構成される分散ファイルシステムのファイル操作方法において、前記クライアントは、前記サーバが持つ階層構造を持たないデータにより構成されるファイル内のデータを、階層構造により表わした階層情報として管理し、該階層情報と前記サーバ内のファイルのデータとの対応情報を保持して、サーバ内のファイルに対する操作が指示されたとき、前記階層情報及び対応情報に基づいて前記サーバのファイルを自クライアントのファイルを操作する方法と同一の方法で操作することを特徴とする分散ファイルシステムのファイル操作方法。

【請求項 5】 請求項 1、2 または 3 記載の分散ファイルシステムを構成するクライアント内の仮想ファイルシステムを実現するプログラムを格納したことを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、分散ファイルシ

ステム及びファイル操作方法に係り、特に、概念の異なるファイルシステムを持つ計算機のファイルをネットワークにより接続して操作することを可能に構成した分散ファイルシステム及びファイル操作方法に関する。

【0002】

【従来の技術】 複数の計算機をネットワークにより接続されて構成されるシステムは、従来、ホストコンピュータや各端末装置の全てが同一のメーカーの製品であることを前提として構成されていたが、近年、ネットワークにおける交換機技術の発達や通信プロトコルの整備により、異なるメーカーのコンピュータを相互に接続してシステムを構築することが可能になっている。

【0003】 前述のようなネットワークシステムを構築する目的は多々あるが、そのうちの 1 つにネットワークのどこか 1 つのマシンにファイルをおき、ネットワークシステムの他のマシンがそれを共用するというファイル共用という目的がある。

【0004】 ネットワークシステムにおいてファイルの共用を行う場合であって、サーバマシンとクライアントマシンとのファイルシステムが異なる場合、そのままではクライアントマシンからサーバマシン上におかれたファイルを取り扱うことができないため、通常、サーバマシンにファイルサーバ機能を入れるなどの方法が取られている。

【0005】 クライアントマシンから異なるファイル管理手段を用いたサーバマシンのファイルを扱う方法に関する従来技術として、例えば、特開平 7-13845 号公報等に記載された技術が知られている。この従来技術は、クライアント・サーバ双方のマシンが所属するネットワークシステムの両方に属する中間のマシンを用意し、中間マシンがクライアントからのサーバファイルに対するアクセス要求を受け付ける機能と、サーバファイルに対してアクセスする機能を持つことにより、クライアントからの要求に従ったサーバファイルの操作を行うことができるようにしたものである。

【0006】 しかし、前述の従来技術による方法は、クライアントが操作したいサーバのファイルを指定することができることを前提としており、クライアントとサーバとが概念的に異なる方法でファイルを管理している場合、クライアントのファイルシステムが、サーバのファイルを特定することができず、アクセス要求自体を行うことができないという問題点がある。

【0007】 例えば、クライアントマシンがパーソナルコンピュータ（以下、PC という）で、サーバマシンがワークステーション（以下、WS という）である場合、双方のファイルシステムは、共に、ファイルとフォルダ（ディレクトリ）による木構造によってファイルを管理するという方式が一般的であり、この場合概念は同一である。このため、クライアント（PC）からサーバ（WS）のファイルを扱う場合でも、木構造を記述してファ

イルを指定するという方法は同一で、クライアントのファイルシステムからサーバのファイルの位置を特定することが可能である。

【0008】しかし、クライアントマシンがPCで、サーバマシンが非UNIX系のベンダのプロプライエタリOSが動作する汎用機である場合、一般的にはサーバマシンのファイルの管理方法がクライアント(PC)とは全く異なることになるため、クライアントのファイルシステムの概念(木構造の記述)によってサーバマシン上のファイルの位置を特定することができない。このため、現状では、サーバマシンのファイルにアクセスするためには、PC上に、汎用機のファイルシステムの概念に従ってサーバのファイルの位置を指定するためのアプリケーションが必要である。この場合、ユーザは、クライアントのファイルとサーバのファイルとを扱うために2つの方法を使い分けなければならない。

【0009】図12は代表的なデータセットの例としての従来技術による区分データセットの構成を示す図であり、以下、この図を参照して、クライアントマシンとサーバマシンとのファイルシステムが概念的に異なる環境である場合の例として、クライアントがPCであり、サーバが汎用機である例について以下に説明する。

【0010】汎用機は、一般に、データセットという概念でデータを取り扱っている。データセットの最小単位はレコードと呼ばれるある特定の長さを持つデータの束である。データセットの構造は、その内部構造であるデータの配列の仕方により異なる。

【0011】区分データセット101は、図12に示すように、1つのデータセットの中がいくつかのメンバと呼ぶ区域102に分割されされており、メンバの名称や位置を管理するディレクトリと呼ばれる領域103がデータセットの先頭に設けられて構成されている。一般に、ディレクトリ領域103は、区分データセット101を操作する際に意識する必要はない。また、利便性のために区分データセット101は、同一の種類のデータの集合、例えば、COBOLソース集合である区分データセットとして作成される。

【0012】データセットは、データセット属性を持つ。データセット属性は、データセットへのアクセス方法や格納しているデータの種類によって決定される。データセット属性は、Data Control Block (DCB) と呼ばれる制御ブロックに格納される。また、データセットは、内部にレコードあるいはメンバと呼ばれるデータの集合を保持することができるが、データセット自体を保持することはない。同様に、メンバもレコードのみを保持することができ、メンバやデータセットを保持することはできない。すなわち、データセットやメンバが階層的な構造になることはない。

【0013】PCのファイル管理は、一般に、ファイルとフォルダという概念が用いられており、レコードとい

う概念はなく、ファイルは可変長のビット列である。フォルダは、ファイルや他のフォルダを含むことができる。このため、PCのフォルダ構造は階層的である。ファイルの持つ属性は汎用機と比較して少なく、特に、ファイルの種類については、ファイル名の後にピリオドで続く文字列(拡張子)を付加する簡易な方法によって管理されている。例えば、XXXという名前のCOBOLソースファイルの場合、“XXX.CBL”のようなファイル名とすることにより便宜的に区別されている。

【0014】操作性に関しては、以下に説明するような違いがある。すなわち、汎用機の操作は、一般に、対話形式の簡易なCUIインタフェースを用いたアプリケーションにより行われているのに対して、PCの操作は、GUIを用いたインタフェースが主流である。このように、PCと汎用機とのファイルシステムは概念的に全く異なっている。

【0015】前述したクライアントがPCでサーバが汎用機である従来技術によるファイル共用システムは、クライアントマシンにサーバの端末機と同等な対話式インタフェースを持つアプリケーションを用意している。そして、ユーザがサーバのファイルにアクセスするために、クライアントのアプリケーションに、サーバである汎用機のファイルシステム用のコマンドを入力すると、クライアントのアプリケーションは、ネットワーク経由でサーバの汎用機のファイルシステムにコマンドを投入する。

【0016】

【発明が解決しようとする課題】前述したように、クライアントがPCでサーバが汎用機である従来技術は、クライアントのアプリケーションの対話式インタフェースが受け付けるのはサーバである汎用機側のファイルシステムのコマンドであるため、クライアントであるPCのユーザが、クライアントのファイルとサーバのファイルとを扱うために全く異なるインタフェースを使い分けることを要求され、このため、作業効率が悪い上、操作ミスが発生しやすいという問題点を有している。

【0017】本発明の目的は、前述した従来技術の問題点を解決し、サーバマシン上のファイルをクライアントのファイルシステムの表示方法により表示し、それに対する操作を受け付けてサーバマシン上のファイル进行处理することを可能とし、これにより、概念が異なるファイルシステムを持つマシン相互間でクライアントのファイル・サーバのファイルを同様に扱うことができるようにし、作業効率を向上させることができ、操作ミスを減少させることができる分散ファイルシステム及びファイル操作方法を提供することにある。

【0018】

【課題を解決するための手段】本発明によれば前記目的は、サーバのファイルシステムと異なるファイルシステムを持つクライアントとサーバとをネットワークにより

接続して構成される分散ファイルシステムにおいて、前記クライアントが、前記サーバが持つ階層構造を持たないデータにより構成されるファイル内のデータを、階層構造により表わした階層情報として管理する階層情報保持手段と、該階層情報保持手段内の情報と前記サーバ内のファイルのデータとの対応情報を保持する対応情報保持手段と、サーバ内のファイルに対する操作が指示されたとき、前記階層情報及び対応情報に基づいて前記サーバのファイルをサーバのファイルシステムを介して操作する操作変換機構とを有する仮想ファイルシステムを備えて構成されることにより達成される。

【0019】また、前記目的は、前記仮想ファイルシステムが、前記操作変換機構がサーバ側ファイルシステムに対する操作を行う際に必要な情報が前記対応情報及び前記階層情報に記述されていない場合、サーバまたはクライアント内を探索することによって前記情報を補完する補完機構をさらに有することにより、また、前記クライアントが、表示機構を備え、該表示装置にクライアントの仮想ファイルシステムが持つサーバのファイルの対応情報を、自クライアントのファイルシステムのファイルと同一の形式で表示させ、クライアントのファイルシステムからサーバのファイルを自クライアントのファイルを操作する方法と同一の方法により操作することにより達成される。

【0020】さらに、前記目的は、サーバのファイルシステムと異なるファイルシステムを持つクライアントとサーバとをネットワークにより接続して構成される分散ファイルシステムのファイル操作方法において、前記クライアントが、前記サーバが持つ階層構造を持たないデータにより構成されるファイル内のデータを、階層構造により表わした階層情報として管理し、該階層情報と前記サーバ内のファイルのデータとの対応情報を保持して、サーバ内のファイルに対する操作が指示されたとき、前記階層情報及び対応情報に基づいて前記サーバのファイルを自クライアントのファイルを操作する方法と同一の方法で操作することにより達成される。

【0021】本発明は、対応情報によってサーバのファイルと関連付けられた階層情報に示される仮想的なファイル情報を表示機構に表示させることにより、サーバのファイルをクライアントのファイルと同様に表示させ、クライアントが、操作変換機構を介してサーバのファイルをあたかもクライアント上のファイルであるかのように、クライアントのファイルの場合と同様に操作することができる。

【0022】また、本発明は、補完機構が、サーバのファイルシステムがファイルを管理するための情報量がクライアントのそれを上回る場合でも、クライアントのファイルシステムと同様の操作をサーバファイルシステムに対するコマンドに変換するために、不足する情報をサーバまたはクライアントに定義されたデフォルト値情報

を検索して取得することができるので、新たなファイルクライアント側からサーバ内に構築することができる。

【0023】

【発明の実施の形態】以下、本発明による分散ファイルシステムの実施形態を図面により詳細に説明する。なお、以下に説明する全ての例において、サーバは非UNIX系ベンダのプロプライエタリOSが動作する汎用機で、取り扱うデータは区分データセットであり、クライアントはPCであるものとする。

10 【0024】図1は本発明の第1の実施形態による分散ファイルシステムの構成例を示すブロック図である。図1において、200はクライアント、201は対応情報保持機構、202は階層情報保持機構、203は表示機構、204は入力受け付け機構、205は操作変換機構、206はクライアントのファイルシステム、207はクライアントのファイル、208はサーバのファイルシステム、209はサーバのファイル、210はサーバ、211は仮想ファイルシステムである。

20 【0025】本発明の第1の実施形態は、PCによるクライアント200と汎用機によるサーバ210とが図示しなすネットワークを介して接続されて構成されている。そして、クライアント200の内部には、クライアントのファイルシステム206と本発明による仮想ファイルシステム211とが備えられ、クライアントのファイルシステム206にはクライアントのファイル207が接続されている。また、サーバ210の内部には、サーバのファイルシステム208が備えられ、サーバのファイルシステム208にはサーバのファイルが接続されている。そして、本発明により設けられたクライアント200内の仮想ファイルシステム211は、対応情報保持機構201、階層情報保持機構202、表示機構203、入力受け付け機構204、操作変換機構205を備えている。なお、以下では、説明の便宜上、201、202はそれらを保持する機構だけでなく情報そのものも指示する符号として使用する。

30 【0026】図1に示す本発明の実施形態は、クライアント200をPCであるとしているので、クライアント200のファイルシステム206は、フォルダ及びファイルによる階層構造によりファイル207を管理している。このため、図示実施形態は、ネットワークを介して多数接続されているサーバ上のデータセットやメンバの中から、クライアント側で参照・更新したいサーバ上のデータセット・メンバに対し、これらをクライアント側で参照するとき、互いを識別するための便宜的な名称を持つ仮想的なファイルを備える仮想ファイルシステムをクライアント200内に用意している。そして、この仮想的なファイルをグループ化するための仮想的な構造として、仮想フォルダを用意する。この仮想フォルダは、他の仮想フォルダ及び仮想ファイルを含むことが可能であるため、仮想フォルダ及び仮想ファイルを用いて階層

構造を表すことができる。この仮想フォルダ、仮想ファイルの名称（サーバ上のデータセット及びメンバのクライアント側での便宜的な名称）及び階層構造を記述する情報を、本発明の実施形態では階層情報と呼び、この情報を階層情報保持機構202の中に保持して、サーバ上のデータセット・メンバによるファイルを管理するために使用する。

【0027】対応情報保持機構201は、階層情報202に記載されている仮想フォルダ及び仮想ファイルと、サーバのデータセット及びメンバとの対応関係を記述する対応情報を保持している。これにより、クライアント200は、サーバのファイル209をクライアントのファイルシステム206と同様の方法で管理することができる。また、表示機構203は、階層情報202に記載されている仮想フォルダ・ファイル及びその関係情報を表示することにより、サーバのファイルの情報をクライアントのファイルシステム206と同様に、仮想フォルダと仮想ファイルとの階層構造として表示することができる。

【0028】図2は仮想ファイル及び仮想フォルダとファイルとの対応の例を説明する図、図3は図2に示す例における階層情報202の記述例を説明する図、図4は図2に示す例における対応情報201の記述例を説明する図である。

【0029】図2において、301は階層情報202に記載されている仮想フォルダ及び仮想ファイルの階層関係の情報を表示機構203に表示したイメージであり、サーバ内の2つの区分データセットのそれぞれを仮想的な階層として表示させたものである。階層情報202は、図3に示すように、仮想フォルダと仮想ファイルとの階層構造を、それぞれの名称と“()”（カッコ）との組み合わせで示すように構成されている。カッコ内の情報は、カッコ外に記載された仮想フォルダの下に位置することを示している。すなわち、階層情報202の一部である“Folder1(File11, File12)”は、Folder1の下にFile11, File12があることを示しており、図2に示す表示イメージ301の中で、仮想フォルダ及び仮想ファイル304～306として表示される構造を示している。

【0030】図1の対応情報201には、階層情報202とサーバのファイル209との対応付けが記載されている。図2に示す対応情報201は、仮想フォルダFolder1304とサーバの区分データセットUSR.DATA1307とを対応させ、仮想ファイルFile11305と区分データセットUSR.DATA1307のメンバMEMBER1308とを対応させている。同様に、対応情報201は、仮想フォルダFolder2309とサーバの区分データセットUSR.DATA2310とを、仮想ファイルFile21311と区分データセットUSR.DATA2310のメンバMEMBER1312とをそれぞれ対応させている。このような対応付けを行う

ことにより、クライアント200は、サーバ側のデータセットをあたかも自クライアント上のファイルであるかのように仮想的に表示することが可能になる。

【0031】図4に示す対応情報201の記述例において、対応情報201は、仮想フォルダ・仮想ファイル名401、対応する汎用機のデータセット・メンバの種別402、対応する汎用機のデータセット・メンバの名前403、属性情報404の組により構成されている。汎用機のデータセット・メンバの種別402は、サーバマシン（汎用機）のデータセット DATASETと、サーバマシン（汎用機）のデータセットのメンバ MEMBER とを含んでいる。汎用機のデータセット・メンバの名前403は、データセット・メンバのサーバ側のファイルシステム上の実際の名称を示す。また、属性情報404は、データセット種別（区分データセットか順データセットか等）、レコード形式（レコードが可変長か固定長か等）等のデータセットの属性情報が記述している。この属性情報404は、仮想フォルダの情報にのみ記載される。仮想フォルダの情報にデータセット名称403及び属性情報404が記載されていない場合、その仮想フォルダにはまだサーバ上のデータセットが対応付けられていないことを示す。

【0032】なお、階層情報202中の仮想フォルダ・仮想ファイルは、クライアントのフォルダ・ファイルを表すこともできる。この場合、クライアントのフォルダやファイルに対して、仮想フォルダ・仮想ファイルの名称と、対応するクライアントのフォルダ・ファイルの名称とを同じにすることによって対応をとる。従って、この場合、図4に示す対応情報201にクライアントのフォルダ及びファイルの情報を記述する必要はない。すなわち、図2に示す例の場合、クライアント（PC）のフォルダFolder3313とPCファイルFile31314とについては、PCフォルダ及びPCファイル名称自体が図3に示す階層情報202内に記述しており、対応情報201内には対応関係情報を記述しない。

【0033】クライアントの仮想フォルダ・ファイルの構造とサーバのデータセット・メンバとは、構造的に一致していなくてもよい。

【0034】図5はこの場合の表示例とサーバのデータセットとの対応情報に基づく対応関係を示す図である。図5において、501は階層情報202に記載されている仮想フォルダ及び仮想ファイルの階層関係の情報を表示機構203に表示したイメージであり、サーバ内の2つの区分データセット1つの仮想的な階層として表示させたものである。

【0035】図5入力示す区分データセット1501と区分データセット2502とは、階層構造ではないが（区分データセットはその内部に区分データセットを持つことはできない）、階層情報としては、それぞれに対応する仮想フォルダFolder11503とFolder150

4とが階層構造になるように保持することができる。

【0036】前述したようにして、PCによるクライアント200は、仮想フォルダ及び仮想ファイルを用いて実体と異なる構造によるデータの管理を行うことが可能になる。

【0037】さて、図1に戻り、入力受け付け機構204は、表示機構203が表示した仮想的なファイル構成（異種のファイルシステムを持つクライアント及びサーバ上のファイルを混在表示する）に対する操作を受け付ける機構である。図2に示している表示の例において、入力受け付け機構204は、仮想フォルダや仮想ファイルに対してマウスによるドラッグ&ドロップ等の操作を受け付ける。また、操作変換機構205は、入力受け付け機構204が受け付けた操作を、クライアントのファイルシステム206またはサーバのファイルシステム208に投入するコマンドに変換する機能を有する。

【0038】図6は操作変換機構の処理動作を説明するフローチャートであり、以下、図6に示すフローを参照して操作変換機構205の処理動作を説明する。

【0039】（1）入力受け付け機構204が受け付けた操作を渡された操作変換機構205は、ユーザの操作対象である仮想ファイル・フォルダに対応する実体を、対応情報から探索する。例えば、ユーザによって、図2に示すFile11 305を削除するという操作が加えられた場合に、操作変換機構は、図4に示す対応情報201を探索する（ステップ601）。

【0040】（2）対応情報を探索した結果、操作対象の仮想フォルダまたは仮想ファイルに対する情報が発見されたか否かにより操作対象がクライアント側のものかサーバ側のものかを判定する。操作対象の仮想フォルダまたは仮想ファイルに対する情報が発見されなかった場合、操作対象がクライアントのファイルかフォルダであると判断できるので、操作変換機構は、クライアントのファイルシステムに投入するコマンドを準備し、そのコマンドを投入する（ステップ602～604）。

（3）ステップ601での対応情報の探索の結果、操作対象の仮想フォルダまたは仮想ファイルの情報が発見された場合、ステップ602で、操作対象がサーバのデータセットまたはメンバであると判断することができ、この場合、操作変換機構は、サーバのファイルシステムに投入するコマンドを準備し、そのコマンドを投入する。図2に示すFile11 305を削除する例の場合、対応情報201を探索した結果、File11 305に対応するものがサーバのデータセットUSR. DATA1 307のメンバMEMBER1 308であることが判るので、操作変換機構は、サーバのファイルシステムに投入するコマンド“DELETE データセット名(メンバ名)”を準備し、サーバのファイルシステムに投入する。すなわち、操作変換機構は、「delete File11」という仮想ファイルに対するコマンドを、「DELETE USR. DATA1 (MEMBER1)」というサーバの

実ファイルに対するコマンドに変換して投入する（ステップ605、606）。

【0041】（4）コマンドの投入によって仮想ファイル・フォルダとクライアントのファイル・フォルダまたはサーバのデータセット・メンバとの対応関係に変更が生じた場合、対応情報を更新する。前述したFile11 305を削除する例の場合、仮想ファイルFile11 305がなくなるので、仮想ファイルFile11 305に対する対応情報を削除する。その後、コマンド投入の結果をユーザに通達し、操作変換機構の処理を終了する（ステップ607、608）。

【0042】前述した本発明の第1の実施形態は、1クライアント・1サーバによる構成を持つものとして説明したが、本発明は、複数クライアント・1サーバによる構成の場合にも適用することができる。この場合、クライアント相互間で共通の対応情報等をネットワーク上に設けて、複数のクライアントがこれらを共有することにより、サーバのファイルを複数のクライアントが参照・更新するようにすることができる。

【0043】図7は本発明の第2の実施形態による分散ファイルシステムの構成例を示すブロック図である。図7において、701は保管機構、702はデフォルト値情報保持機構であり、他の符号は図1の場合と同一である。

【0044】図7に示す本発明の第2の実施形態は、図1に示した本発明の第1の実施形態に対して、クライアント200の仮想ファイルシステム211内に補完機構701が追加され、この補完機構にデフォルト値を与えるデフォルト値情報保持機構702が接続されて構成されている。

【0045】この本発明の第2の実施形態は、サーバのデータセットやメンバを新規に作成する場合等に使用して好適なものである。例えば、ユーザの操作の結果として、新しいデータセットが作成される場合、新規データセットを割り当てるコマンドALLOCATEが属性の情報を要求する。しかし、図1に示す本発明の第1の実施形態における対応情報201には、新規データセットに設定する属性の情報が記述されていないため、操作変換機構204は、サーバのファイルシステム208に投入するコマンドを準備することができなくなる。本発明の第2の実施形態は、このような状況でも操作をサーバ側に伝達することができる。

【0046】図7において、操作変換機構205は、本発明の第1の実施形態の場合と同様に、入力変換機構204に対する操作の対象を対応情報201から探索し、操作対象を発見できた場合、操作の対象がサーバのデータセット及びメンバに対するものであれば、サーバのファイルシステム208に投入するコマンドを準備する。このとき、対応情報201及び階層情報202の情報のみではコマンドを準備できない場合、操作変換機構20

5は、補完機構701を呼び出す。補完機構701は、コマンドの決定に必要な情報をクライアントやサーバ内から探して、操作変換機構205に戻す。

【0047】なお、前述において、階層情報202の情報のみではコマンドを準備できない場合としては、新規データセットを割り付ける場合等がある。新規データセット割り付けのコマンドには、属性の情報を指定する必要があるが、対応情報201には、既存のデータセットの属性しか記載されていないため、操作変換機構は、新規データセットの属性が判らず、構成のような場合にコマンドを準備することができない。

【0048】図8はサーバ上に新規に区分データセットを作成して、そこにクライアント(PC)のフォルダFolder2の下にあるCOBOLソースファイル“File21.CBL”をコピーする動作例を説明する図である。このとき、新規の区分データセットに対して、仮想フォルダFolder1は、事前に対応付けられているものとする。

【0049】まず、ユーザは、File21.CBL 803に対応する仮想ファイル804をマウスでドラッグ&ドロップするなどして、仮想フォルダFolder1 801の下にコピーする操作を行う。このとき、操作変換機構205は、コピー先の仮想フォルダ801の情報を対応情報201から発見し、サーバのデータセットに対応する仮想フォルダであることを知る。

【0050】ここで、操作変換機構205は、サーバのファイルシステム208に投入するコマンドを準備する。この例では、その詳細を後述するが、ALLOCATEコマンドとCOPYコマンドとが準備される。ALLOCATEコマンドは、データセット名を指定することにより、新規に指定された名称のデータセットを割り当てるコマンドである。このとき、属性情報を指定することにより新規に作成したデータセットに属性を指定することができる。また、COPYコマンドは、指定されたメンバ名称のメンバをデータセット内に新規に作成し、その内容として、ネットワークを通じて別に指定したクライアント側のファイルの内容を複写するコマンドである。

【0051】図9は操作変換機構の処理動作を説明するフローチャート、図10は補完機構の処理動作を説明するフローチャートであり、以下、図9、図10に示すフローを参照して操作変換機構205の処理動作を説明する。

【0052】(1) 入力受け付け機構204が受け付けた操作を渡された操作変換機構205は、ユーザの操作対象である仮想ファイル・フォルダに対応する実体を、対応情報201から探索し、操作対象がクライアントのフォルダ・ファイルであるか、サーバのデータセット・メンバであるかを、操作対象の情報が対応情報に存在するかないかにより判定する。なお、判定の結果、操作対象がクライアント側のファイルやフォルダであった場合の処理は、図6により説明したと同様であるので、こ

こでの説明を省略する(ステップ901、902)。

【0053】(2) ステップ902の判定で、操作対象がサーバのデータセット・メンバであった場合、操作変換機構205は、ユーザの操作に対応するコマンドを準備するが、前述したような新規データセットの割り当て時のように、このコマンドに指定するパラメータが対応情報201及び階層情報202から決定できない場合、補完機構701を呼び出す処理を開始する。このため、まず、操作対象のデータセットが既存であるかどうかを判定する(ステップ903)。

【0054】(3) 説明している例では、操作対象のデータセットは既存のものではないとしているので、ALLOCATEコマンドを準備するために、ALLOCATEコマンドに指定するデータセットの属性のパターンを決定する(ステップ904)。

【0055】(4) ここでは、コピー元のファイルFile21.CBL 803が COBOLソースファイルであることから、新規に作成するデータセットに対して COBOLソースファイルに好適な属性で与えて良いという判断を行う。そして、新規に作成されるデータセット名称及びメンバ名称を、仮想フォルダ801及び仮想ファイル804の名称から決定する。なお、仮想フォルダ名からデータセット名称を決定するロジック、及び、仮想ファイル名からメンバ名称を決定するロジックの例については後述する(ステップ905)。

【0056】(5) ここまでのステップによる処理で、データセット名称、メンバ名称を決定することができたが、指定する属性は、ステップ904で COBOLソースファイルに好適なものを用いるということを決めたのみで、実際の値は、仮想ファイルシステム内の情報から決定することができないため、操作変換機構は補完機構701を呼び出す(ステップ906、907)。

【0057】(6) 呼び出された補完機構701は、ステップ904で決定したデータセット属性のパターンに従って、クライアントマシンに置かれている予め定義されたデータセット用属性のデフォルト値情報をデフォルト値情報保持手段702を探索し、不足している情報、ここでは COBOLソースファイル用データセットの属性を取得する。なお、属性のデフォルト値情報保持機構702は、サーバ側に配置しておいて、サーバ側でデフォルト値を検索してもよい(ステップ1001)。

【0058】(7) その後、補完機構701は、情報獲得の成否を判定し、成功の場合、獲得した情報を操作変換機構に戻し、また、失敗の場合、情報を獲得することができなかったことをユーザに伝達する(ステップ1002～1004)。

【0059】(8) 操作変換機構205は、得られた情報を元にコマンドを準備する。図8に示す例の場合、準備するコマンドは、“ALLOCATE データセット名 属性情報”と“COPY PCファイル名 データセット名(メンバ

名)”とである。そして、操作変換機構205は、このコマンドを図7に示すサーバのファイルシステム208に投入する(ステップ908、909)。

【0060】(9)コマンドの投入の結果、仮想ファイル・フォルダとサーバのデータセット・メンバとの対応に変更があった場合、操作変換機構205は、対応情報の更新を行い、最後に、操作完了をユーザに通知して処理を終了する(ステップ910、911)。

【0061】なお、前述では、属性情報をクライアントに問い合わせるとして説明したが、本発明は、対話的インタフェースを用いてユーザに入力させるようにすることもできる。

【0062】前述したステップ905での処理における、仮想フォルダ名からデータセット名称を決定するロジック及び仮想ファイル名からメンバ名称を決定するロジックの例について以下に説明する。

【0063】仮想フォルダ名からデータセット名称を決定するロジック

(1)仮想フォルダ名内の英小文字をすべて英大文字に変換する。

(2)メンバ名称として使用不可能な文字が仮想フォルダ名中にある場合、予め用意した規則(“!”→“1”等)に従って使用可能な文字に変換する。

(3)ユーザ名称の後に、ピリオドを挟んでメンバ名称を接続する。例えば、ユーザ名が“USR”の場合、“USR.”の後に(2)で作成したメンバ名称を接続する。一般に、汎用機にログオンする際にユーザ名を入力する機会が多いため、ここでは仮想ファイルシステムがユーザ名を既に知っているものとする。ユーザ名が未知の場合、何らかのデフォルト値を使用しても良い。

【0064】仮想ファイル名からメンバ名称を決定するロジック

(1)仮想ファイル名の拡張子の前の“.”(ピリオド)を外す。

(2)仮想ファイル名内の英小文字をすべて英大文字に変換する。

(3)メンバ名称として使用不可能な文字がファイル名中にある場合、予め用意した規則にしたがって使用可能な文字に変換する。

【0065】前述したロジックを図8に示す例に適用した場合、仮想フォルダFolder1901の名称から新規作成されるデータセットの名称は、ユーザ名を“USR”であるとしたとき“USR.FOLDER1”となり、仮想ファイルFile21.CBLの名称から新規に作成されるメンバの名称は、“USR.FOLDER1(FILE21)”となる。

【0066】なお、前述した本発明の実施形態は、仮想フォルダ・ファイル名からデータセット名称及びメンバ名称をロジックによって決定するとして説明したが、本発明は、自動的に決定せずにユーザにデータセット名称・メンバ名称を入力させる等の他の方法を用いるように

することもできる。

【0067】図11は図8に示す例での更新前後の対応情報の例を示す図である。図11において、1101は操作前の対応情報を示しており、1102は操作後の対応情報を示している。

【0068】前述した図10に示す補完機構の処理において、情報を得られなかった場合、操作変換機構は、ステップ1004での処理でユーザにその旨を通知して処理を終了しているが、ユーザに入力を求める等により処理の継続を試みるようにすることもできる。

【0069】前述した本発明の実施形態は、前述したクライアントでの各種の処理手順を実行するプログラムを、FD、CDROM、MO、DVD等の記憶媒体に格納して提供することができる。

【0070】前述した本発明の実施形態によれば、クライアントにおいて、サーバマシン上のファイルをクライアントのファイルシステムの表示方法で表示し、それに対する操作を受け付けてサーバマシン上のファイルを実行することが可能になる。これにより、概念が異なるファイルシステムを持つマシン間でもクライアントのファイル・サーバのファイルを同様に扱うことができ、作業効率を向上させ、クライアントのファイル・サーバのファイルを別々の方法で取り扱う場合に比べて操作ミスを減少させることができる。

【0071】という効果がある。

【0072】

【発明の効果】以上説明したように本発明によれば、概念が異なるファイルシステムを持つマシン相互間でクライアントのファイル・サーバのファイルの両者を同様に扱うことができ、作業効率を向上させ、操作ミスを減少させることができる。

【図面の簡単な説明】

【図1】本発明の第1の実施形態による分散ファイルシステムの構成例を示すブロック図である。

【図2】仮想ファイル及び仮想フォルダとファイルとの対応の例を説明する図である。

【図3】図2に示す例における階層情報の記述例を説明する図である。

【図4】図2に示す例における対応情報の記述例を説明する図である。

【図5】表示例とサーバのデータセットとの対応情報に基づく対応関係を示す図である。

【図6】操作変換機構の処理動作を説明するフローチャートである。

【図7】本発明の第2の実施形態による分散ファイルシステムの構成例を示すブロック図である。

【図8】サーバ上に新規に区分データセットを作成して、そこにクライアントのフォルダの下にあるCOBOLソースファイルをコピーする本発明の第2の実施形態の動作例を説明する図である。

【図 9】 操作変換機構の処理動作を説明するフローチャートである。

【図 10】 補完機構の処理動作を説明するフローチャートである。

【図 11】 図 8 に示す例での更新前後の対応情報の例を示す図である。

【図 12】 代表的なデータセットの例としての従来技術による区分データセットの構成を示す図である。

【符号の説明】

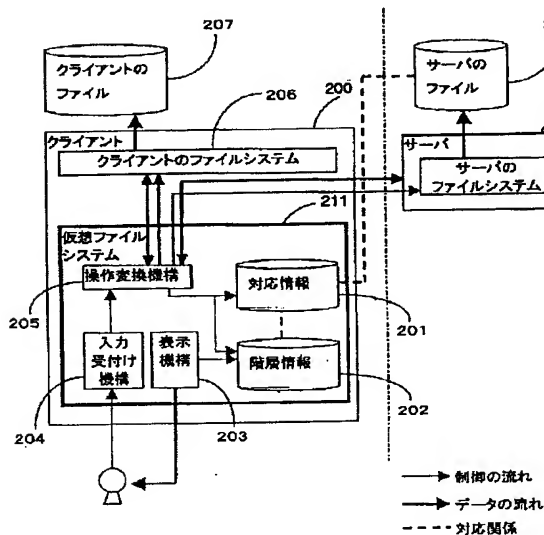
200 クライアント
201 対応情報保持機構
202 階層情報保持機構

203 表示機構
204 入力受け機構
205 操作変換機構
206 クライアントのファイルシステム
207 クライアントのファイル
208 サーバのファイルシステム
209 サーバのファイル
210 サーバ
211 仮想ファイルシステム
701 保管機構
702 デフォルト値情報保持機構

10

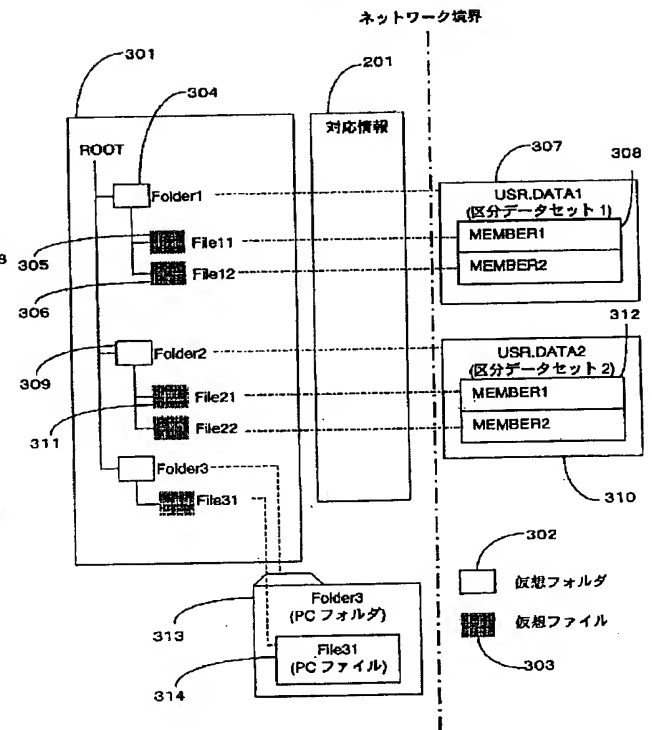
【図 1】

図 1



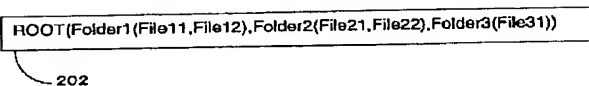
【図 2】

図 2



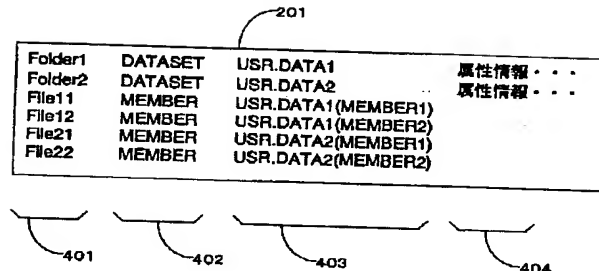
【図 3】

図 3



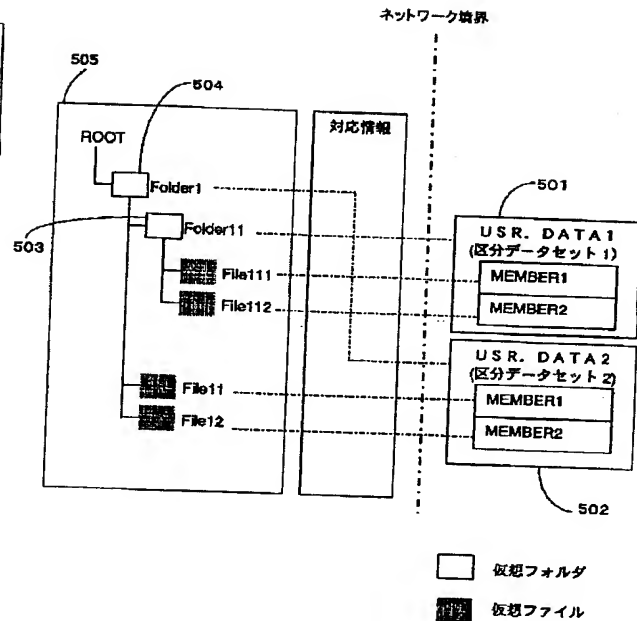
【図4】

図4



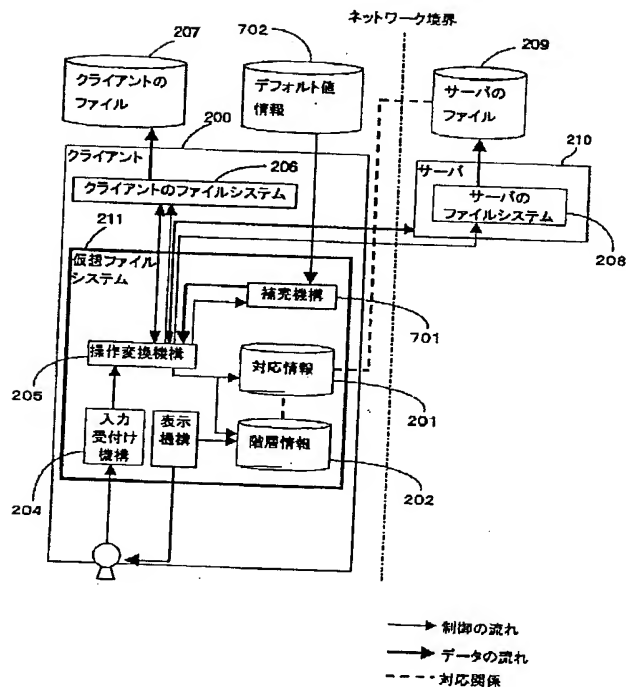
【図5】

図5



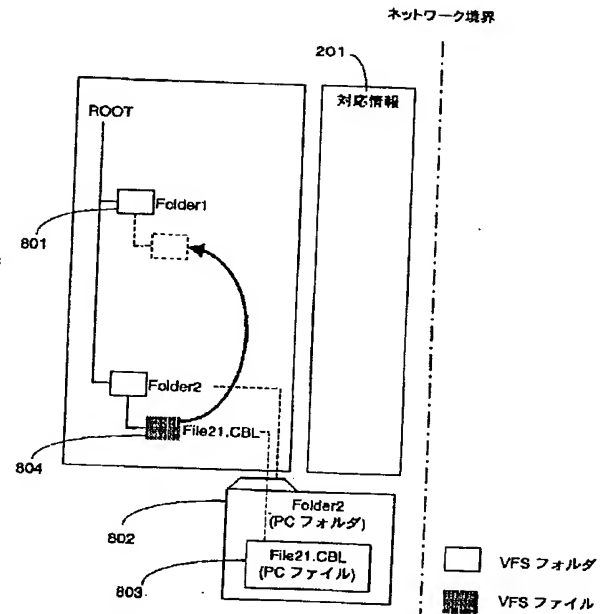
【図7】

図7



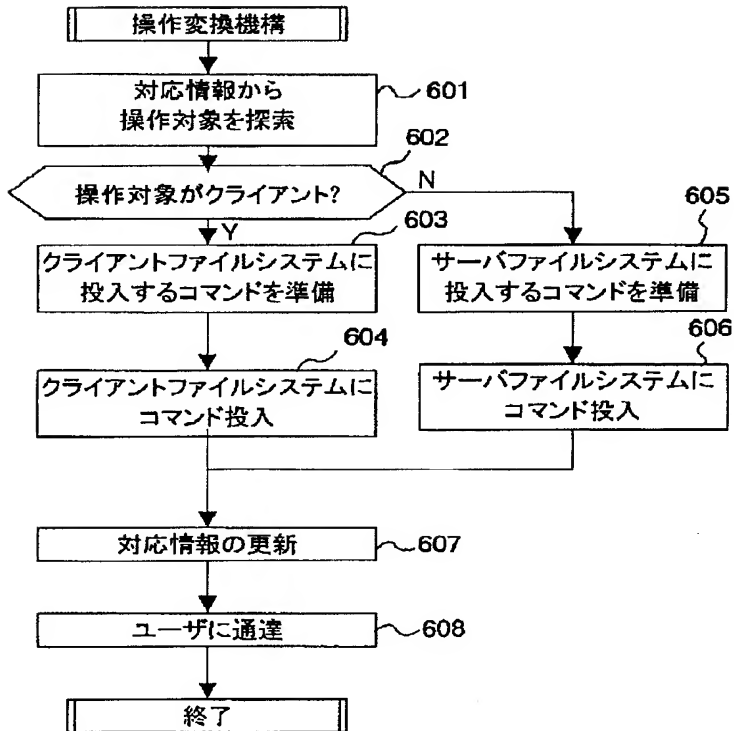
【図8】

図8



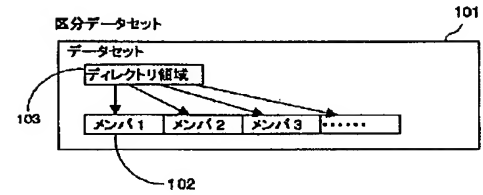
【図6】

図6



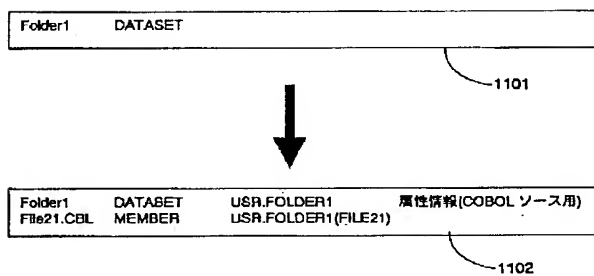
【図12】

図12



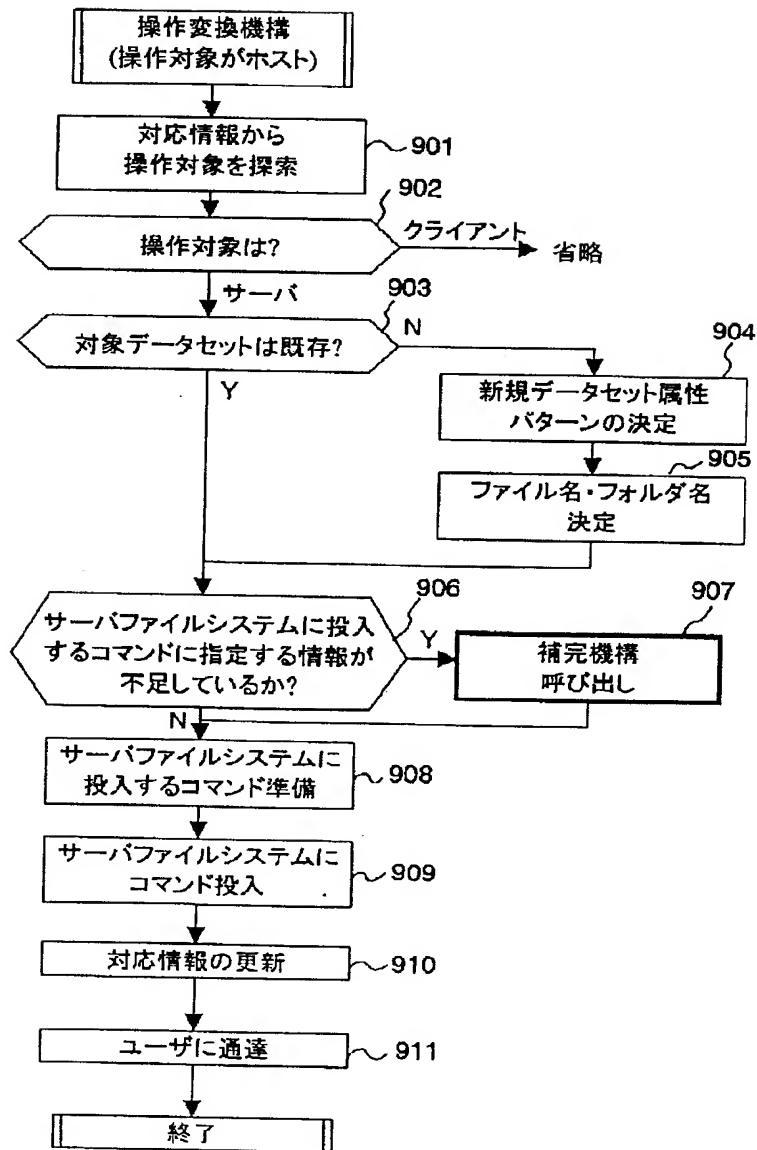
【図11】

図11



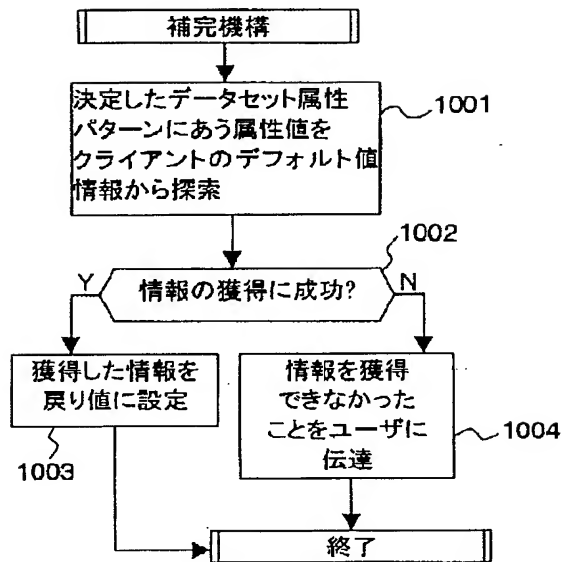
【図9】

図9



【図10】

図10



フロントページの続き

(72) 発明者 河井 政雄
神奈川県横浜市戸塚区戸塚町5030番地 株
式会社日立製作所ソフトウェア事業部内

(72) 発明者 茨木 暢靖
神奈川県横浜市戸塚区品濃町549-6 株
式会社日本コンピュータ研究所内
Fターム(参考) 5B082 EA01 EA07 GC03

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-188702

(43)Date of publication of application : 10.07.2001

(51)Int.Cl.

G06F 12/00

(21)Application number : 11-373771

(71)Applicant : HITACHI LTD

(22)Date of filing : 28.12.1999

(72)Inventor : NUMANOI ATSUSHI
YOKOZUKA ONORI
KAWAI MASAO
IBARAKI NOBUYASU

(54) DISTRIBUTED FILE SYSTEM AND FILE OPERATING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To improve working efficiency, and to reduce any operation miss by dealing both the file of a client and the file of a server between machines which are provided with file systems whose concepts are different.

SOLUTION: This distributed file system is provided with a hierarchical information 202 for operating convenient designation for virtually treating the file of a server as the file of a client, and for defining the hierarchical structure of the convenient mutual designations and corresponding information 201 for describing the corresponding relation of the convenient designation with an actual file 209 of the server. Also, this distributed file system is provided with a display mechanism 203 for displaying the information of the files by referring to the corresponding information 201 and the hierarchical mechanism 202, an input accepting mechanism 204 for accepting the same operation as that of the file system of the client for the file of the server, and an operation converting mechanism 205 for converting the operation into an operation corresponding to the file system of the server.

